# Nexikon Society

## (Society Management System)

Rani Kumari, Satyam Kumar, Nithun Kumar, Raghav Talwar, Shivam Kumar Chaudhary
*BCA(BigDataAnalytics)*
*LovelyProfessionalUniversity*
Jalandhar,Punjab
rani.12103298@lpu.in, satyam.1211070@lpu.in,nithun.12115999@lpu.in,raghav.12106936@lpu.in

***Abstract*— The "The Nexikon Society" is an offers an integrated platform aimed at facilitating effective communication, information sharing, and issue resolution within residential communities. This platform serves as a centralized hub where society members can access real-time updates, notices, and event announcements, fostering a stronger sense of community engagement and collaboration.**

**Through the Nexikon website, residents have easy access to various community-related information, including updates on ongoing projects, upcoming events, and important notices. Additionally, the platform allows members to view profiles of their neighbour's, facilitating networking and social interactions within the community.**

**A significant feature of Nexikon is its complaint management system, which enables members to report issues or concerns pertaining to the society. This feature ensures that grievances are addressed promptly and promotes transparency and accountability among society administrators.**

**Moreover, the website restricts database modification privileges to the administrator, ensuring the integrity and security of the data. The administrator is tasked with managing member information, overseeing platform operations, and performing essential tasks to ensure its smooth functioning.**

Keywords—
***Society Management System, Residential Communities, Communication, Information Dissemination, Issue Resolution, Administrator, Database Management.***

**Introduction-**

Welcome to our Society Management System Website Nexikon, a comprehensive platform crafted to modernize our community management and improve the well-being of all our members.

In today's rapidly evolving world, conventional approaches to managing society affairs often prove inadequate, leading to inefficiencies, communication gaps, and unresolved issues. Our system aims to address these challenges by offering an intuitive, centralized solution tailored to meet the diverse needs of our community.

A standout feature of our system is the introduction of a virtual notice board accessible through our website. This digital platform ensures that all members can stay informed about vital announcements, events, and updates, regardless of their location or schedule.

To uphold the integrity and security of our system, we have implemented two distinct logins: one for administrators responsible for managing society affairs and another for members to access pertinent information and services. This dual-login system not only safeguards the authenticity of information but also promotes transparency and accountability within our community.

Our system goes beyond just providing a means of communication; it also prioritizes addressing member concerns and grievances promptly. Through our complaint register feature, members can raise issues related to water, cleanliness, technical works, or any other concerns they may have. Our streamlined process ensures that these complaints are acknowledged, tracked, and resolved efficiently, fostering a sense of trust and satisfaction among our members.

By transitioning to a paperless environment, we're not only lessening our environmental impact but also

simplifying administrative tasks and reducing paperwork for both members and management. Our centralized database houses all pertinent information, such as member details, event schedules, and complaint records, ensuring secure and easy access to manage data.

Our Society Management System represents more than just a tool—it's a catalyst for positive change within our community. Join us as we embrace this digital shift and collaborate to establish a more efficient, transparent, and cohesive living environment for all our members. Together, we can forge a brighter future for our society.

**1.Requirement Specification**

This requirement specification delineates the essential software, hardware, and data requisites for the Society Management System. By addressing these elements, the system endeavours to offer a user-friendly, secure, and effective platform for overseeing society affairs and fostering communication among members.

**1. Software Requirements:**

- **Operating System:** The system is compatible with Windows operating systems.
- **Web Browser:** Any web browser can be used, with preference given to Google Chrome for optimal performance.
- **Server Software:** XAMPP or WAMP server should be installed on the user's system to facilitate local hosting and testing.

**2. Hardware Requirements:**

- **Server:** A server with a minimum of 2 GB of storage space is required for hosting the system.
- **RAM:** The server should have at least 4 GB of RAM to ensure smooth operation.

**3. Data Requirements:**

- **Username:** Each user must have a unique username to access the system.
- **Password:** A secure password is required to protect user accounts.
- **Email ID:** Users need to provide an email address for communication purposes.
- **Mobile Number:** Each user must register a mobile number for important notifications and alerts.
- **Flat Number:** Users should specify their flat numbers for identification within the society.

**2. Fact-Finding Questions:**

- **Overall System Functionality:** What are the core tasks and features provided by the system?
- **Admin Roles:** How many administrators are allowed, and what are their respective responsibilities?
- **System Management:** Who will oversee the day-to-day operations and maintenance of the system?
- **Data Security:** How does the system ensure the confidentiality and integrity of user data?
- **Membership Limit:** Is there a maximum limit to the number of members who can register within the society?
- **Updating Information:** How can users modify their personal details within the system?
- **Admin User Access**: Do administrators have regular user access or special privileges?
- **Support and Queries:** Where can users seek assistance or raise concerns about the system?

# 3. System Design Details

## *3.1 Event Table:*

**Database Setup:**

- Our approach involves structuring our data into tables for various purposes such as user accounts, complaints, notices, etc., ensuring efficient information management.
- Employing techniques like indexing and normalization, we aim to enhance data

accessibility, making it easier to locate and retrieve relevant information.

**Backend Work:**

- Our system's core functionality will include managing user logins, complaint submissions, and other essential tasks.
- To streamline development, we'll utilize frameworks like Django or MYSQL to handle web requests and manage server actions effectively.

**Frontend Development:**

- This stage involves creating the user interface components that users interact with, including registration and login pages, the notice board, and complaint submission forms.
- We'll utilize web technologies such as HTML, CSS, and JavaScript, potentially incorporating frameworks like React or Bootstrap to enhance the appearance and functionality of the interface, ensuring a seamless user experience.

**Security Measures:**

• Ensuring the safety of our system, we'll employ robust encryption to safeguard user passwords, implement secure connections (HTTPS), and fortify defenses against common threats like SQL injection attacks.

**Scalability and Speed:**

• Our system architecture will be engineered to support high user traffic without compromising performance and enhancing the functionality of the system.

• To optimize speed, we'll implement strategies such as caching (storing frequently accessed data for rapid retrieval) and distributing the workload across multiple servers to maintain efficient operations even during peak usage periods.

**Monitoring and Keeping Things Running Smoothly:**

• We'll establish monitoring systems to continuously assess the performance of our system, enabling early detection of any potential issues.

• Through regular updates and maintenance, we'll uphold system functionality, ensuring security and incorporating the latest features to keep our system current and efficient.

## *3.2 Class Diagram:*

The class diagram for a society management website illustrates the interaction among various components such as User, Admin, Notice, Complaint, Authentication, Dashboard, Notification, and Database, outlining their roles in providing functionality. For example, User interacts with Authentication during the login process, while Admin interacts with Complaint to handle user complaints. This diagram visually depicts the structure and interconnection of different parts of the website, demonstrating how they collaborate to deliver the desired functionality.

1. **User Class:**
   - This component represents the individuals utilizing the website. It encompasses details like their username, password, and contact information.
2. **Admin Class:**
   - Like the User class this component specifically pertains to administrators with have special permissions. They have the authority to manage users, handle address complaints, and perform other administrative etc.
3. **Notice Class:**
   - This class manages the notices published on the website, encompassing announcements or event details. It comprises attributes like the notice content and posting date of the users for easier access.
4. **Complaint Class:**

   - This class oversees the complaints submitted by users, encompassing attributes such as the complaint content, status (resolved or pending), and submission date.
5. **Authentication Class:**
   - This class manages the process of user login and authentication process, ensuring that only registered users can gain access the system.
6. **Dashboard Class:**
   - This component represents the primary page or interface users encounter upon logging in. It grants access to various features like viewing notices, submitting complaints, etc.
7. **Notification Class:**
   - This will handle notifications sent to users, which may include email alerts for new notices or updates on complaint statuses.

## 4. Future Enhancement

- To improve the project further, consider integrating online maintenance payment functionality. This would automate payment processing and generate receipts for every member, reducing the administrative workload.
- Implement message and email alerts to keep users informed about various events and updates within the society. This will ensure that members don't miss important information.
- Introduce parking management features to address parking issues within the society premises.
- Implement a guest registration system for every flat to enhance security and monitor visitor entry effectively.
- Incorporate online meeting attendance capabilities to facilitate virtual meetings, reducing the need for physical gatherings and promoting convenience for all members.
- Implement a facility booking system to allow residents to reserve common areas such as the clubhouse, gym, or community hall for events or gatherings.
- Introduce a suggestion box feature where residents can submit their ideas and feedback anonymously to improve community living.
- Develop a lost and found platform where residents can report lost items and others can return them if found, fostering a sense of community responsibility.
- Integrate a community directory with contact information of all residents, making it easier for neighbours to connect and communicate.
- Implement a complaint management system to address issues or grievances raised by residents in a timely manner, ensuring efficient resolution.
- Introduce a community events calendar to keep residents informed about upcoming social gatherings, maintenance schedules, or other important dates.
- Implement a green initiative tracker to monitor and encourage environmentally friendly practices within the community, such as waste recycling or energy conservation efforts.

## 5. Conclusion

In this phase of the pandemic, it's crucial to have systems in place, like a society management system, to keep our communities running smoothly.

This project aims to streamline tasks such as managing records of society members, distributing notices to everyone, and handling complaints.

Imagine having all the information about who lives in your building or neighbourhood neatly organized in one place. No more sifting through piles of papers or trying to remember who lives where. Plus, sending out notices about important events or changes becomes a breeze, ensuring everyone stays informed.

And when it comes to addressing issues or concerns, this system makes it easy for residents to

submit complaints or requests for maintenance, making sure they're heard and addressed promptly.

Overall, it's about making life easier for everyone in the community by cutting down on manual work and keeping communication flowing smoothly, especially during these challenging times.

# 6. System Design Details

| No. | Event | Trigger | Source | Activity | Response | Destination |
|---|---|---|---|---|---|---|
| 1. | Register now | formcontainer | User | Creates new account | Adds new account to db, redirect to home | Server |
| 2. | Login | formcontainer | User | Checks username and password to login | Redirect to home | Server |
| 3. | Admin | btn | User | Login to admin page | Redirect to admin | Server |
| 4. | User wants to view notice board | View notice | User | Click on to view notice button | Notice section | Local |
| 5. | User wants to submit complaint | Btn-Complaint | User | Submit complaint of user | Add complaint to db, redirect to home | Server |
| 6. | User wants to upload photos | Btn-upload | User,admin | Click to upload photo | To view photo gallery | Server |
| 7. | Logout | Btn logout | User | Logout from account | Redirect to login page | Server |
| 8. | Add member | Btn-add | Admin | Click to add newMember | Redirect to manage members | Server |
| 9. | Update member | Btn-update | Admin | Click to update member | Redirect to manage members | Server |
| 10. | Delete member | Btn-Delete | Admin | Click to delete member. | Redirect to manage members | Server |
| 11. | Add notice | Btn-notice | Admin | Click to add notice. | Redirect to add notice | Server |
| 12. | View complaints | Btncomplaints | Admin | Click to view complaints | Redirect to View Complaints | Server |
| 13. | Logout | Btn-Logout | Admin | Click to logout | Rdirect to admin login page | Server |

# 7. Results

## Validation and Naming Conventions:

| Sr.No | Control ID | Validation Used | Reason |
|---|---|---|---|
| 1. | Username | RequiredFieldValidator | Username Cannot be null |
| 2. | Email | RequiredFieldValidator | Email Cannot be Null |
| 3. | Flat no. | RequiredFieldValidator | Flat no. Cannot be null |
| 4. | Mobile no. | RequiredFieldValidator | Mobile no. Cannot be null |
| 5. | No. of family members | RequiredFieldValidator | No.of family Members cannot be null |
| 6. | Password | RequiredFieldValidator | Password cannot be null |
| 7. | Title | RequiredFieldValidator | Title cannot be null |
| 8. | Subject | RequiredFieldValidator | Subject cannot be null |
| 9. | Notice type | RequiredFieldValidator | Notice Type cannot be null |

# 8. References

w3schools

tutorials point.

javatpoint

YouTube